Toward Affordance-Aware Planning

David Abel & Gabriel Barth-Maron, James MacGlashan, Stefanie Tellex
Dept. of Computer Science, Brown University

RSS: First Workshop on Affordances [July 13, 2014]

Goal

Enable autonomous agents to learn how to plan efficiently in massive stochastic state spaces.

Affordances

Affordances: knowledge added to an MDP that directs the agent toward relevant action possibilities.

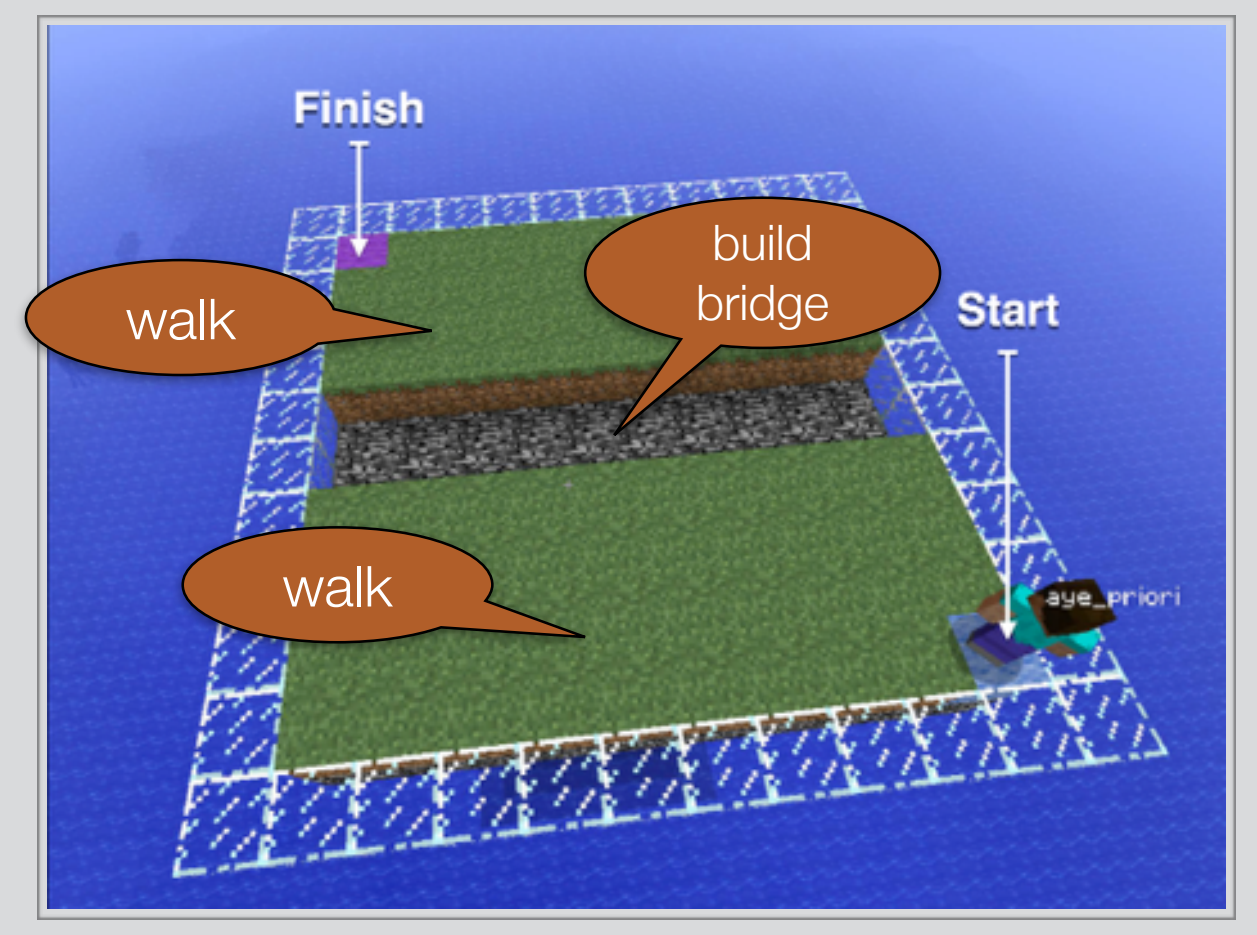
Formalism:

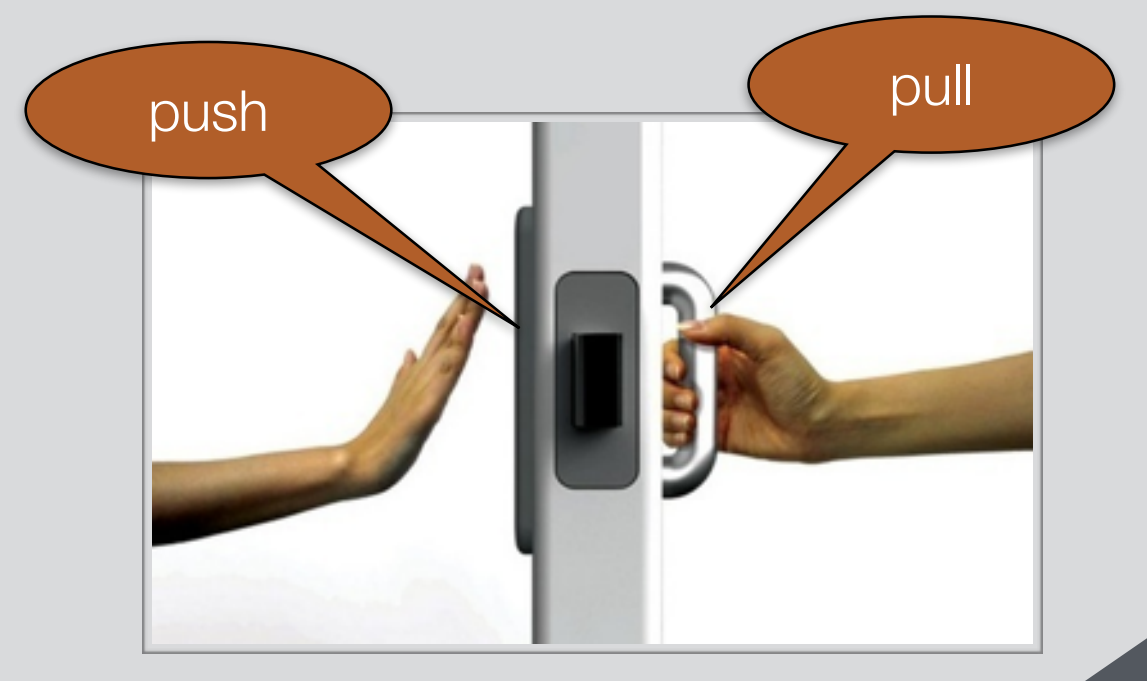
$$\Delta = \langle p, g \rangle \longmapsto \mathcal{A}'$$

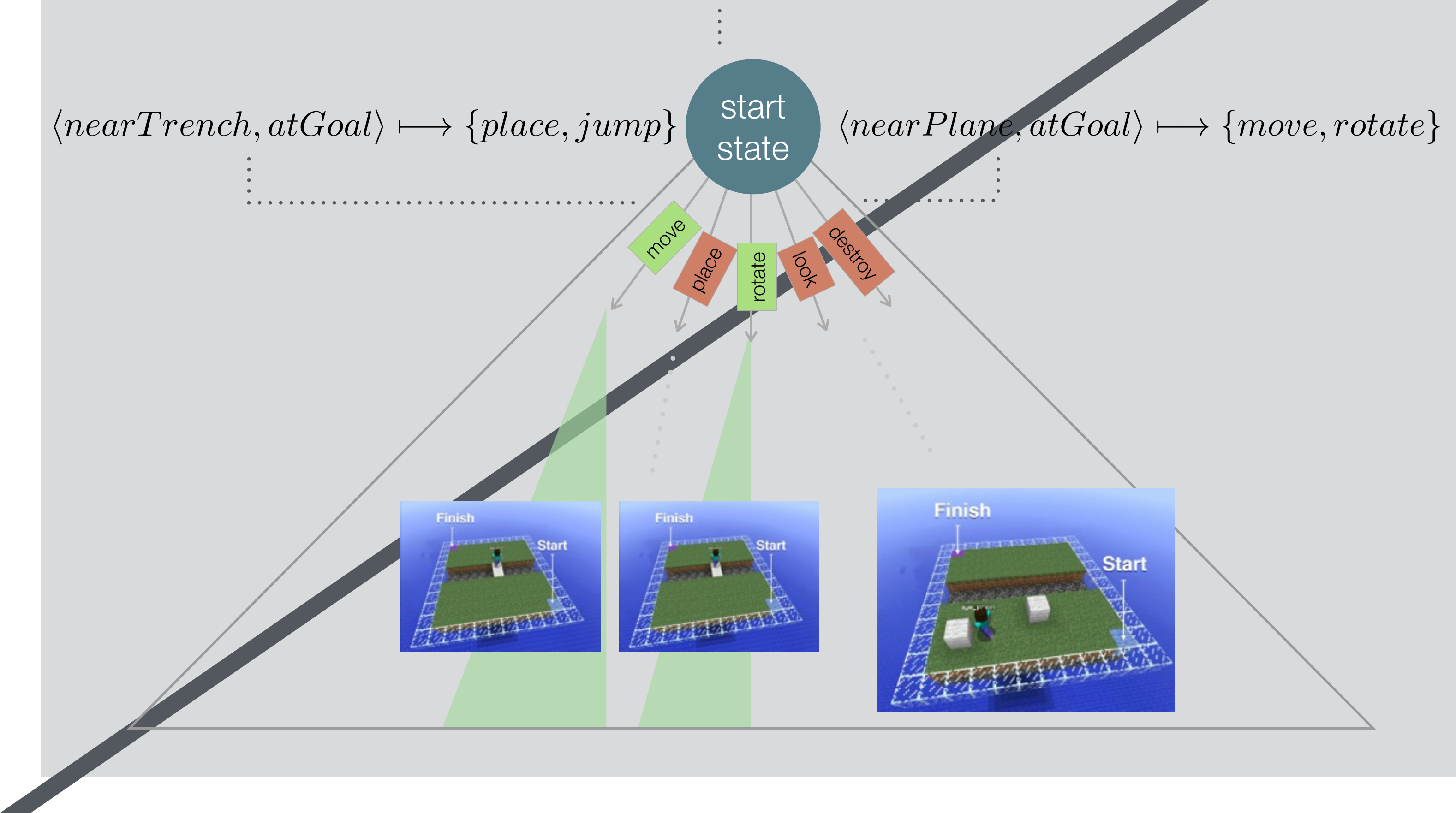
Where:

Δ = symbol for an affordance
p = predicate on states
g = lifted goal description
$\mathcal{A}'$ = subset of MDP Actions

Example: Affordances in Minecraft



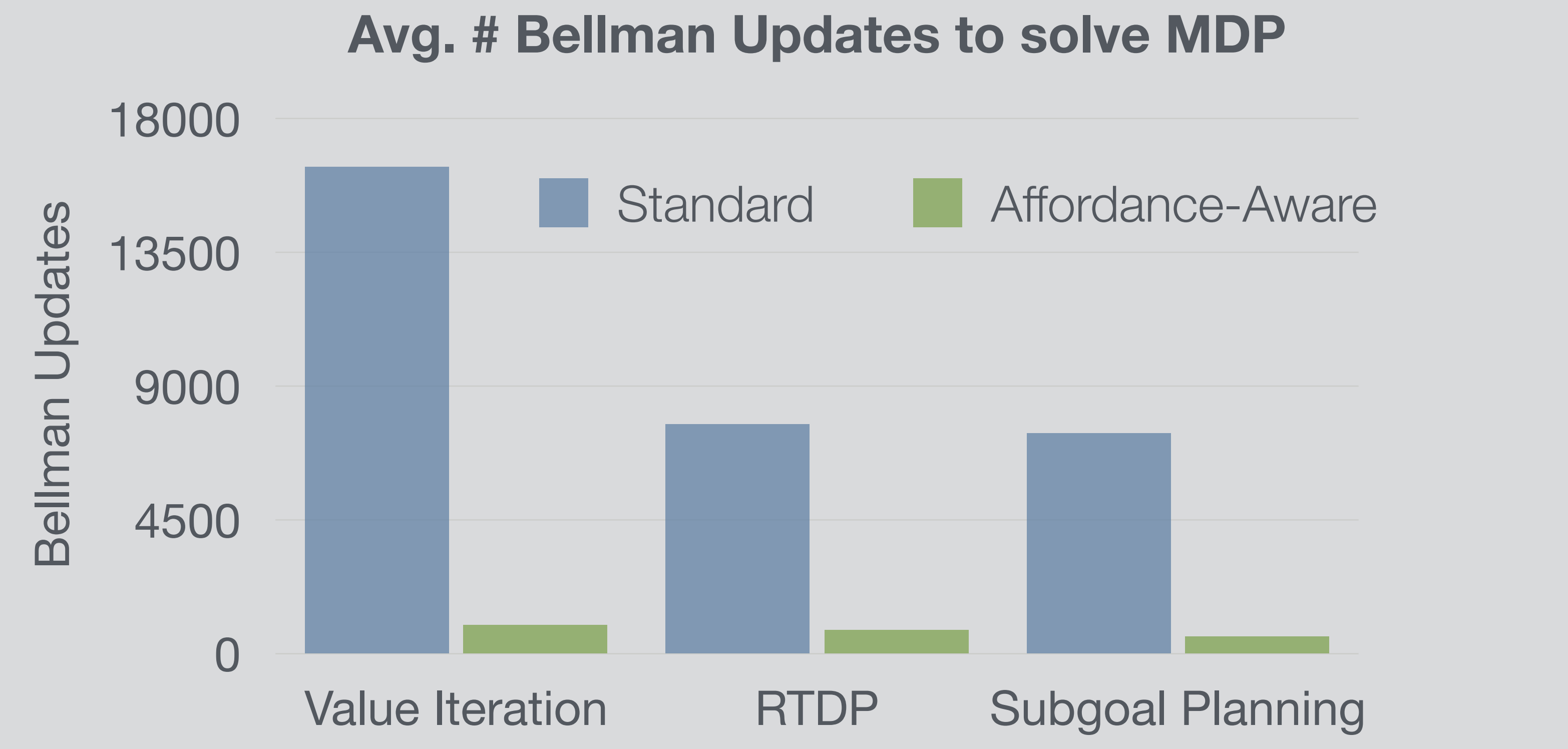




$\langle nearTrench, atGoal \rangle \longmapsto \{place, jump\}$

$\langle nearPlane, atGoal \rangle \longmapsto \{move, rotate\}$

Preliminary Results



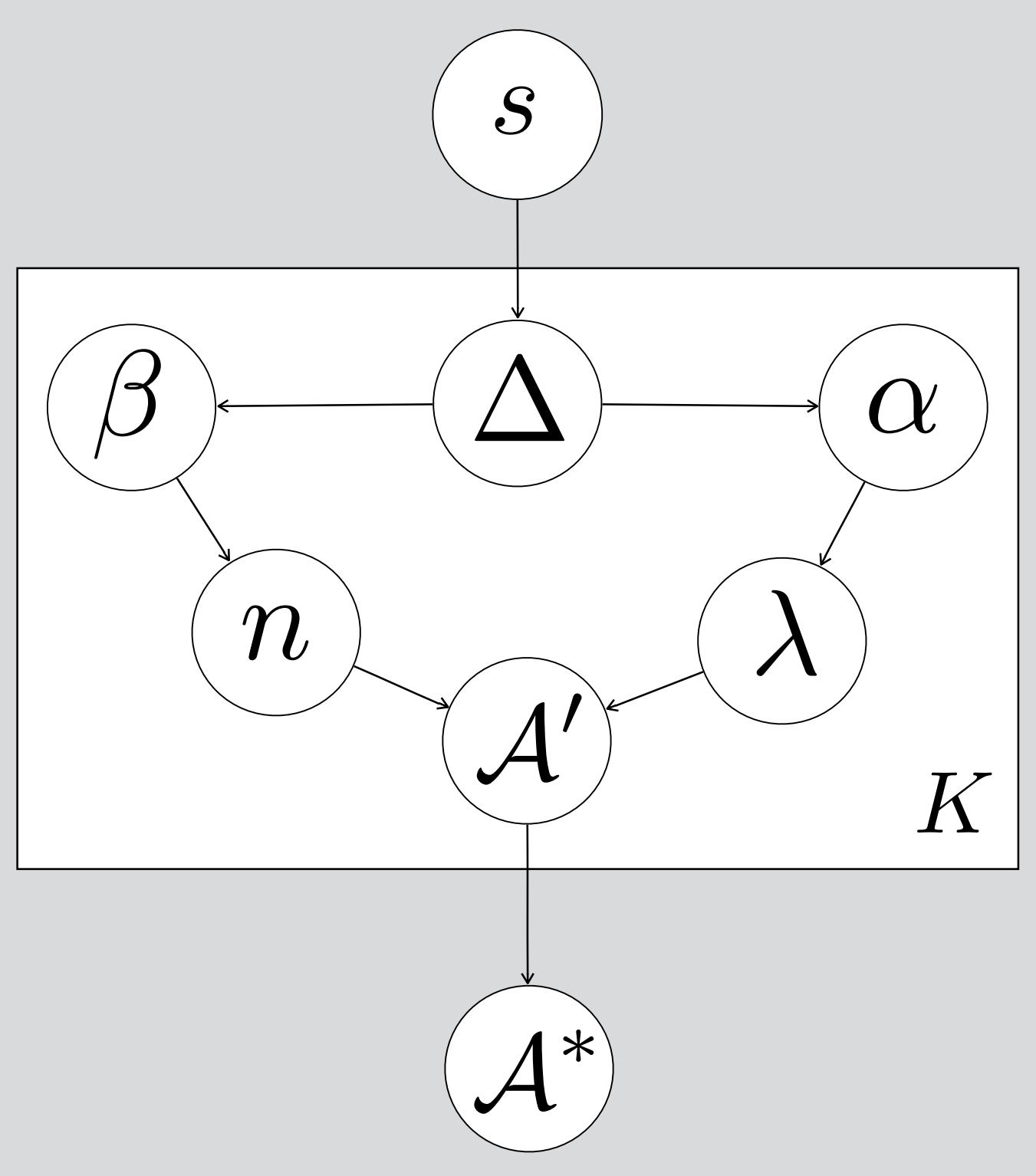
Learning

Goal: For a given state *s*, learn which actions are most relevant

$$\Pr(\mathcal{A}^* \mid s, \Delta_1 \ldots \Delta_K)$$

$$= \Pr(\mathcal{A}'_1 \cup \ldots \cup \mathcal{A}'_K \mid s, \Delta_1, \ldots, \Delta_K) \approx \sum_{i}^{K} \Pr(\mathcal{A}'_i \mid s, \Delta_i)$$

Graphical Model:



s = OO-MDP State

Δ = Affordance

α = Action Counts

β = Action Set Size Counts

λ = Multinomial on Actions

n = Multinomial on Action Set Size

$\mathcal{A}'$ = An Affordance's Action Set

$\mathcal{A}^* = \bigcup_{i=1}^{K} \mathcal{A}'_i$

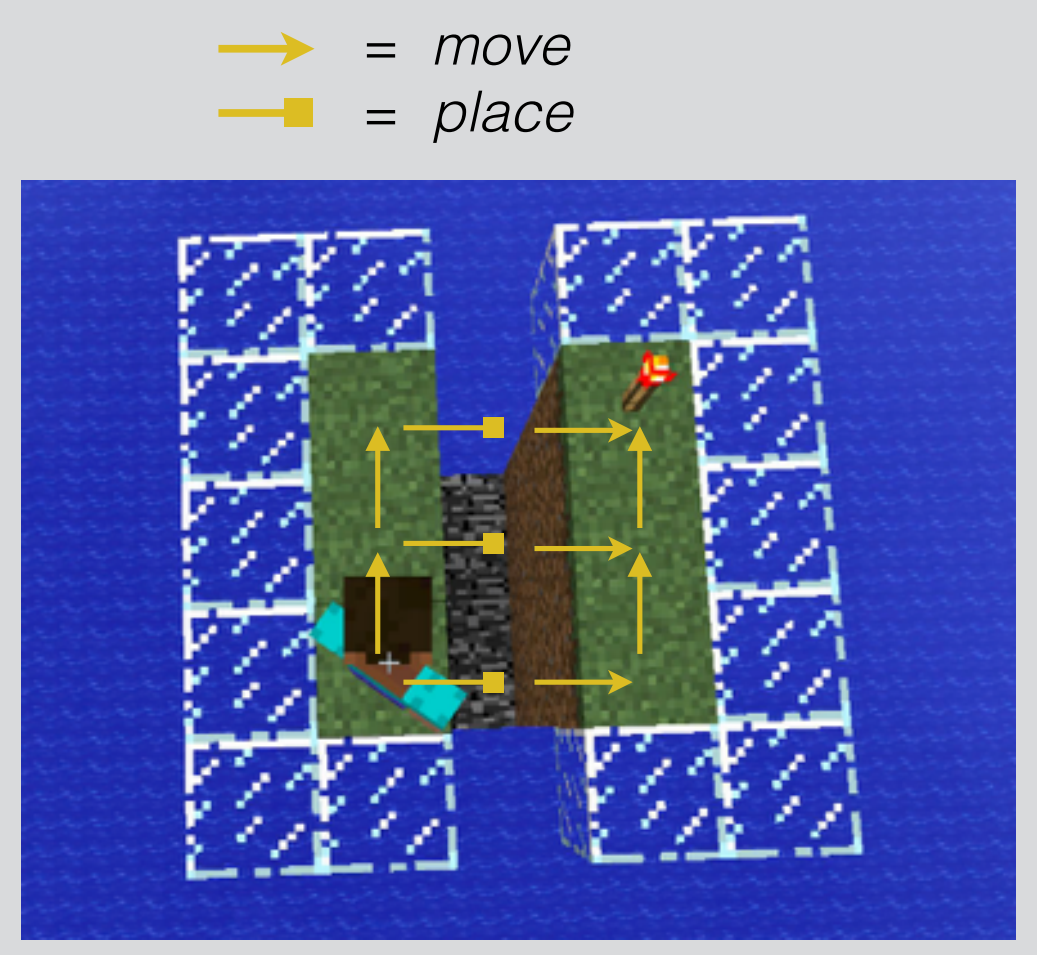
$$\Pr(\lambda_i \mid \alpha_i) = DirMult(\alpha_i) \qquad \Pr(n_i \mid \beta_i) = Dir(\beta_i)$$

Learning Example

1) For each activated affordance, count:

$\alpha =$ number of worlds in which each action was used

$\beta =$ number of unique actions used in each world



$\Delta_1 = \langle ✓ \, nearTrench, ✓ \, atGoal \rangle$

$\Delta_1.\alpha.moveRight$++, $\Delta_1.\alpha.moveForward$++, $\Delta_1.\alpha.placeRight$++

$\Delta_1.\beta.3$++

2) When solving the MDP on a new state space, in each state *s*:

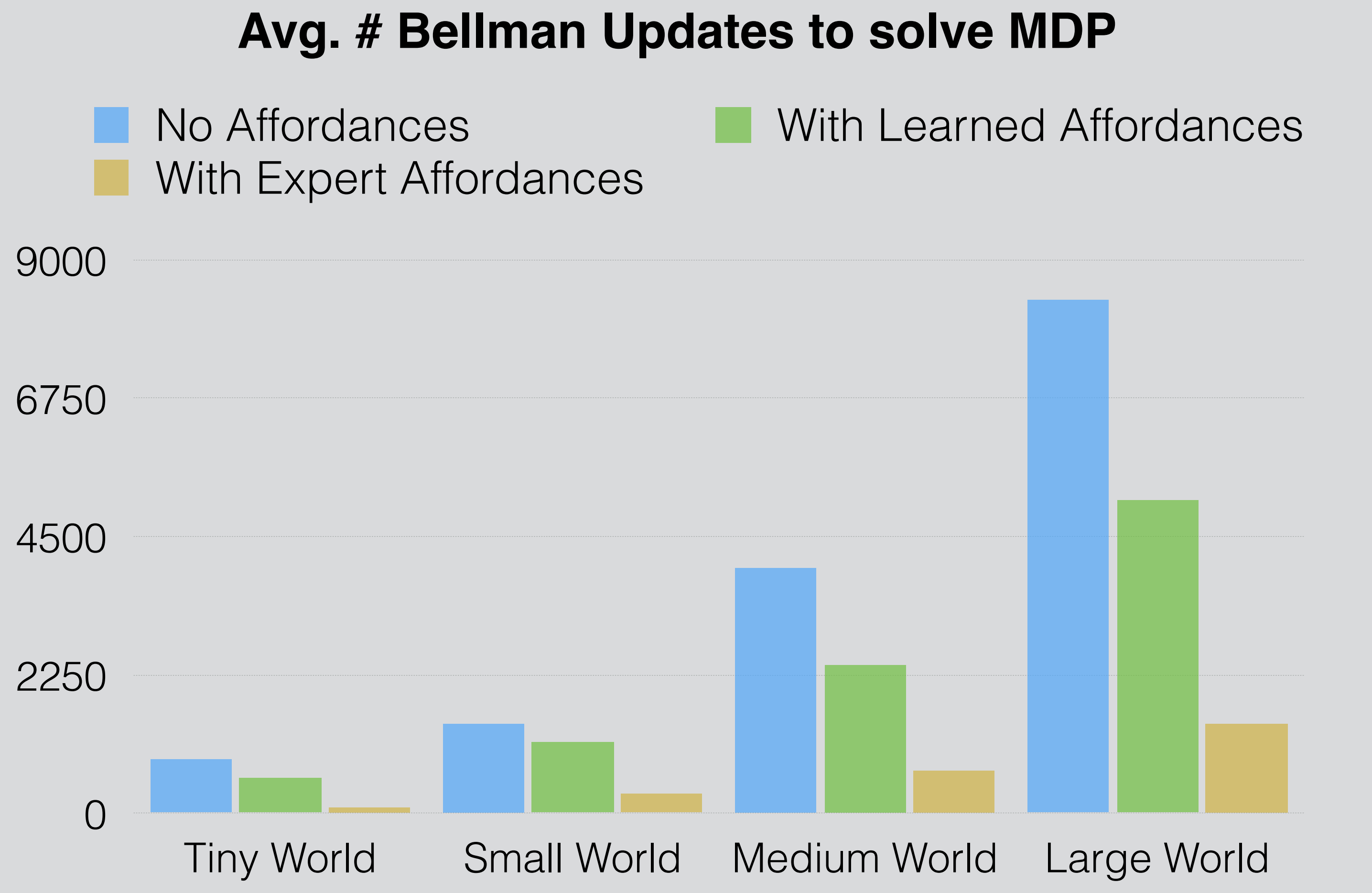
$$\mathcal{A}^* = \bigcup_{i=1}^{K} (\Delta_i.getActions(s))$$

3) Where

$\Delta_i.getActions(s)$:

```
$\lambda \leftarrow DirMult(\Delta_i.\alpha)$
$n \leftarrow Dir(\Delta_i.\beta)$
$\mathcal{A}' \leftarrow_n \lambda$
return: $\mathcal{A}'$
```

Learning Results


Avg. # Bellman Updates to solve MDP
No Affordances
With Learned Affordances
With Expert Affordances
Tiny World
Small World
Medium World
Large World